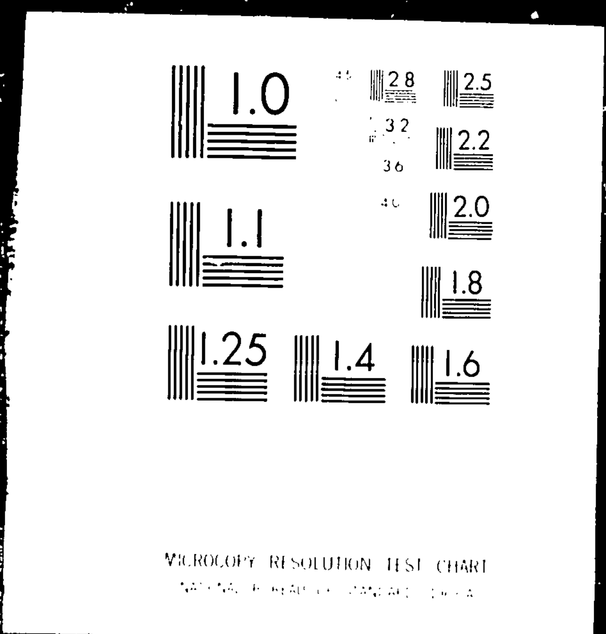
ED 06

7 883

1 OF 1

ED 067 883



DOCUMENT RESUME

ED 067 883 EM 010 374

AUTHOR	Stelzer, John; Garneau, Jean
TITLE	Project IMPACT Software Documentation: Overview of the Computer-Administered Instruction Subsystem.
INSTITUTION	Human Resources Research Organization, Alexandria, Va.
SPONS AGENCY	Office of the Chief of Research and Development (Army), Washington, D.C.
REPORT NO	HumRRO-TR-72-21
PUB DATE	Aug 72
NOTE	36p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Assisted Instruction; *Computer Programs; Computer Science; Flow Charts; Information Processing; *Military Training; *Program Descriptions; Time Sharing
IDENTIFIERS	*Project IMPACT; U S Army

ABSTRACT

Research in Project IMPACT, prototypes of computerized training for Army personnel, is documented in an overview of the IMPACT computer software system for computer-administered instruction, exclusive of instructional software. The overview description provides a basis for an understanding of the rationale and motivation for the development of the IMPACT computer software. In a general way, it covers the support software for time-sharing, authoring, data management, and instructional decision modeling. Flow charts are provided to show the general pattern of information processing within the system. Each of the components is described in technical detail in a series of separately available research products. (Author)

Technical Report 72-21

HumRRO-TR-72-21

ED 067883

HumRRO

Project IMPACT Software Documentation: Overview of the Computer-Administered Instruction Subsystem

John Stelzer and Jean Garneau

HUMAN RESOURCES RESEARCH ORGANIZATION
300 North Washington Street • Alexandria, Virginia 22314

August 1972

Prepared for
Office of the Chief of Research and Development
Department of the Army
Washington, D.C. 20310

Destroy this report when it is no longer needed.
Do not return it to the originator.

ED 067883

HumRRO
Technical
Report
72-21

Project IMPACT Software Documentation: Overview of the Computer-Administered Instruction Subsystem

John Stelzer and Jean Garneau

HumRRO Division No. 1 (System Operations)
Alexandria, Virginia

HUMAN RESOURCES RESEARCH ORGANIZATION

Work Unit IMPACT

August 1972

Approved for public release; distribution unlimited.

Prepared for

Office of the Chief of Research and Development
Department of the Army
Washington, D.C. 20310

The Human Resources Research Organization (HumRRO) is a nonprofit corporation established in 1969 to conduct research in the field of training and education. It is a continuation of The George Washington University Human Resources Research Office. HumRRO's general purpose is to improve human performance, particularly in organizational settings, through behavioral and social science research, development, and consultation. HumRRO's mission in work performed under contract with the Department of the Army is to conduct research in the fields of training, motivation, and leadership.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Published
August 1972
by
HUMAN RESOURCES RESEARCH ORGANIZATION
300 North Washington Street
Alexandria, Virginia 22314

Distributed under the authority of the
Chief of Research and Development
Department of the Army
Washington, D.C. 20310

FOREWORD

This report presents an overview of the first-generation software products of Project IMPACT, Prototypes of Computerized Training for Army Personnel. These products will be described in detail in a series of research documents (available to specialists in the field through information retrieval sources) dealing with the support software for time-sharing, authoring, data management, and instructional decision modeling. Work Unit IMPACT is an advanced development project undertaken by the Human Resources Research Organization to provide the Department of the Army with a computer-administered instruction (CAI) system. The National Science Foundation is also sponsoring HumRRO research on Instructional Decision Models (IDMs), with additional support provided by the James McKeen Cattell Fund.

The research is being conducted at HumRRO Division No. 1 (System Operations) Alexandria, Virginia, where Dr. J. Daniel Lyons is Director. Dr. Robert J. Seidel is the Program Director. Technical contributions were made by Dr. John Stelzer and Mr. Jean A. Garneau.

Earlier work in the same general area as the IMPACT Project was conducted under Work Unit METHOD, Research for Programed Instruction in Military Training, and Exploratory Study 42, Organization of Instruction. Some of the principal publications under these research efforts include: *Project IMPACT—Computer-Administered Instruction: Preparing and Managing the Content of Instruction, IMPACT Text-Handling Subsystem*, Technical Report 71-21, September 1971; *Software Documentation Series, Project IMPACT—Computer-Administered Instruction: Functions for the Coursewriter III Language*, Research Product RBP-D1-71-2, June 1971; *Project IMPACT—Computer-Administered Instruction: Description of the Hardware/Software Subsystem*, Technical Report 70-22, December 1970; and *Project IMPACT: Computer-Administered Instruction Concepts and Initial Development*, Technical Report 69-3, March 1969.

Identification of products is for research documentation purposes only, and does not constitute an official endorsement by HumRRO, the Department of the Army, the National Science Foundation, or the James McKeen Cattell Fund.

HumRRO research for the Department of the Army is conducted under Contract DAHC 19-73-C-0004. Computer-Administered Instruction research is conducted under Army Project 2Q063101D734. The IDM research being conducted under National Science Foundation sponsorship is funded under Grant GJ-774, Research on Instructional Decision Models, with additional funds from the James McKeen Cattell Fund.

Meredith P. Crawford
President
Human Resources Research Organization

SUMMARY AND CONCLUSIONS

MILITARY PROBLEM

The combination of shrinking financial resources and the prospects of a smaller, all-volunteer Army will increase both the demands made on Army personnel and the importance of the individual soldier. There will be a greater need for more effective and efficient training, adequate to the task of providing an increasing number of complex skills to widely differing students, while using fewer skilled instructors.

The most promising approach available to meet these new training demands is computer-administered instruction (CAI), if it is developed as a comprehensive, total system.

The goal of Project IMPACT is to provide the Army with an effective, efficient, and economical CAI system in a total system framework. To be effective, the system should maximize the achievement of the students and the instructors to a greater extent than is possible in the traditional classroom; to be efficient, it should provide maximum productivity per unit time on the part of instructors, administrators, and students; and to be economical, the cost and resources must not exceed those of a comparable effective non-CAI instructional system.

DEVELOPMENT PROBLEM AND APPROACH

Project IMPACT was established by the Department of the Army in 1968 as an advanced development effort to provide a total system of CAI for the effective and efficient training of military personnel. Accordingly, a Technical Development Plan (TDP) was conceived that provides for the concurrent development of the four facets of a total CAI system: instructional content, hardware, software, and instructional decision model (IDM). The Project was organized to keep these facets in balance over a span of two generations of CAI systems and four successive cycles of development and testing. The initial two cycles covering the development and test of a "breadboard" CAI system have been completed. The second two cycles were planned as a period for refinement of all facets of the system, to produce a prototype model to be tested, evaluated, and then delivered to the Army as specifications for an operational instructional system.

In pursuing its goal, Project IMPACT has followed an evolutionary approach toward developing products usable by Army instructional staff. This document describes the overall first generation, IMPACT-A, software products. The intent for widespread Army use is to provide functional requirements for a cost/effective system.

PRODUCTS

The documents in the software series have been prepared to assist systems programmers in incorporating all or some of the IMPACT-A software products into on-going CAI efforts. While the primary purpose of the initial generation was to develop and test a provisional total CAI system, many of its products, such as the time-sharing software, data management capabilities, and IDM guidelines can fulfill user needs now. Subsequent products from the continued effort (IMPACT-B) will document the revision and refinements to these items. The software products are:

(1) Zeus Documentation—operationally available time-sharing software; authoring command set; separate text and course logic facility.

(2) FACS (File Activity Control System)—a set of computer programs that provides information concerning display pages that are disk stored; a system to assist in editing and coordinating displays.

(3) IDES-1 (IMPACT Data Evaluation System—Version 1)—a set of computer programs that manage the data collected, stored, and processed by the CAI system (no longer used).

(4) IDES-2 (IMPACT Data Evaluation System—Version 2)—an updated software system that provides for storage, retrieval, and analysis of student generated data.

(5) The Interface—a system that maintains on-line records of the prerequisites that have been satisfied by each student; it also controls intermodule and intramodule transfers.

(6) Coursewriter III Functions—a version of IBM's Coursewriter III that performs response analyses, data generation, and branching for students and data manipulation capabilities.

An overview of this software system is presented in this report. The products are described in detail in a series of Research Products intended primarily for personnel working in the computer software field.

CONTENTS

Section		Page
1	Introduction	3
2	Project IMPACT	4
	Background	4
	Objectives of Hardware/Software Subsystem	4
	Approach to the Development of Hardware/Software Subsystem	5
	CAI Activities Supported by Subsystem	5
	Hardware Components in Subsystem	5
3	Overview of IMPACT's Software System	7
	Real-Time vs. Off-line Operations	7
	IMPACT's Real-Time Software System	7
	Terminals	9
	The Operating System (OS)	10
	Zeus	10
	Coursewriter	10
	Instructional Text and Instructional Logic	11
	Discussion	11
	IMPACT's Off-line Software System	12
	The Operating System	12
	IDES	12
	FACS	12
4	System Logic	15
	Real-Time System Logic	15
	Off-line System Logic	21
5	System Requirements	23
6	Conclusions	24
Literature Cited		27

Figures

1	IMPACT's Current Hardware System	6
2	Functional Architecture for IMPACT's Real-Time System	8
3	IMPACT's Real-Time Software Components	9
4	IMPACT's Off-line Software Components	13
5	Overview of Real-Time System Logic	16
6	Input to the Real-Time System	17
7	Real-Time System Logic	18
8	Off-line System Logic	22
9	Minimum Core Requirements for System Components	23

Project IMPACT Software Documentation: Overview of the Computer-Administered Instruction Subsystem

Section 1

INTRODUCTION

This report presents an overview of the Project IMPACT computer software system for computer-administered instruction (CAI), *exclusive of instructional software*. The distinction between computer and instructional software is between programs that drive the computer and programs that actually administer instruction.

In a series of detailed Research Products, the CAI software research and development effort conducted in Project IMPACT over the past four years will be described. The series is intended primarily for programmers who might be responsible for implementing and maintaining the computer software system on their own computers. However, nontechnical persons, such as administrators, working in the CAI area will be able to use this overview report to gain an understanding of the rationale and motivation for the development of the IMPACT computer software.

Although the computer software is referred to in these documents as a system, it should be emphasized that it is actually a *subsystem* of the overall IMPACT instructional system. These documents will not deal directly with the other main research and development efforts conducted under Project IMPACT—Instructional Decision Modeling and CAI hardware. These efforts have been described and documented in other reports (1, 2). A general introduction to Project IMPACT is included in the present report but will not be repeated in the Research Product series.

Other documents have dealt with various aspects of the software system, including the main components in the hardware/software system (3); the Coursewriter III[1] functions developed in IMPACT (4); and the text-handling subsystem that is composed of authors, software, hardware, and established procedures (5). This new series of software documents is intended to complement and extend the previous documentation efforts. The IMPACT software system will be described as it is presently configured; however, since the software research and development effort is ongoing, at least minor changes can be expected to occur over time.

HumRRO will provide installations interested in using the IMPACT system with versions of the total software system or of individual components. The Research Products in the series will provide the precise details needed for implementing, using, and maintaining the IMPACT software. These items, available through information retrieval systems, include the following:

Project IMPACT Software Documentation

- II. The IMPACT Data Evaluation System—Version 2
- III. The IMPACT Data Evaluation System—Version 1
- IV. The Interface Subsystem Framework for Instructional Decision Modeling
- V. File Activity Control System (FACS)
- VI. Volume 1, Zeus Functions and Design Concepts
  Volume 2, Zeus Program Logic Descriptions
- VII. The IMPACT Computer-Administered Instruction Software Subsystem, Coursewriter III, and Its Functions
- VIII. Computer-Administered Instruction Computer Program Logic for COBOL2 Course of Instruction

[1] For brevity, Coursewriter III will be referred to as Coursewriter.

Section 2

PROJECT IMPACT

BACKGROUND

The use of general purpose digital computers for instruction is a relatively recent development in educational technology. "Computer-*assisted* instruction," as it is generally called, holds vast promise for providing an instructional system that is more effective, efficient, and economical than present, traditional systems and methods. Its promise lies in the potential of the computer for adapting instruction to the momentary needs and capabilities of the individual student, for use at his own pace. Project IMPACT, Prototypes of Computerized Training for Army Personnel, was established as an integrated, multidisciplinary CAI effort. The objective of the effort is to evolve, through cyclical development and evaluation, an effective, efficient, and economical computer-administered instructional system. The term "computer-*administered*" is used because the computer in this system houses the controlling and adaptive instruction decision model (IDM).

Problem areas addressed by IMPACT include computer system capabilities, CAI language needs and potential and, of prime importance, the formulation of instructional strategies and their relationships to learning processes. Project IMPACT is unique in that it considers the total instructional system. The problem for any instructional agent (human or machine) is to take optimal action in line with an overall "best" strategy for transmitting to the student information uniquely relevant to him. If proficiency criteria are to be attained effectively and efficiently, recurrent decisions concerning these instructional actions must be made relative to (a) the subject matter being taught, (b) the specific student, (c) the momentary circumstances, and (d) the available options (communication channels).

OBJECTIVES OF HARDWARE/SOFTWARE SUBSYSTEM

The objectives of the IMPACT CAI hardware/software subsystem are, first, to provide, within the development time and cost constraints, a generalized and flexible tool for development of CAI instructional decision models and courses of instruction and second, to provide computer hardware and software that is operationally efficient and economical for large-scale CAI use. The hardware/software subsystem is designed to provide the flexibility needed to develop courses in a variety of subjects of instruction, according to a wide range of instructional strategies, for a spectrum of student populations, within a framework of development and experimentation. The hardware/software prototypes must, therefore, provide a balanced capability that will meet the requirements for instructional development (needed to develop the IDM and course prototypes), and the requirements for efficient large-scale instructional operations.

APPROACH TO THE DEVELOPMENT OF HARDWARE/SOFTWARE SUBSYSTEM

The approach taken toward the development of the hardware/software subsystem reflects a compromise between immediate and longer-range requirements. As part of the IMPACT Technical Development Plan,[1] "off-the-shelf" hardware and software components were used in the first generation system wherever possible, and tailored or modified to provide greater development flexibility and potential for operational efficiency and economy. The approach used to achieve the balance between development and operational requirements was to design the tools needed for instructional system development and to implement these tools and capabilities in computer hardware and software technology that can be expanded and refined to provide operational efficiency.

CAI ACTIVITIES SUPPORTED BY SUBSYSTEM

The IMPACT hardware/software subsystem supports the following four main activities in the CAI environment:

(1) Presenting instruction to students.
(2) Implementing instructional material into CAI format.
(3) Evaluating students, courses, and instructional decision model.
(4) Performing administrative functions.

HARDWARE COMPONENTS IN SUBSYSTEM

In support of these activities, the subsystem includes the following main hardware components:

(1) An IBM 370 Model 145 central processor with associated mass, random-access storage, and sequential storage devices.[2]
(2) Nine Sanders 720 Cathode Ray Tubes (CRTs) with keyboards for information displaying and student inputting. Each CRT also has a lightpen.
(3) Student/computer communication devices including random-access film projectors, and experimental devices such as a tablet for handwritten student input and a voice recognition device for audio student input.

The IMPACT hardware system is summarized in Figure 1.

An introduction to the IMPACT computer software system is presented in the following three sections of this report. An overview of the system is presented in Section 3, in Section 4 the system logic is described, and in Section 5 the software system prerequisites are described.

[1] A Project IMPACT Technical Development Plan on instructional model prototypes attainable in computerized training was produced in December 1966.

[2] Identification of products is for research documentation purposes only and does not constitute an official endorsement by HumRRO or the Department of the Army.

IMPACT's Current Hardware System

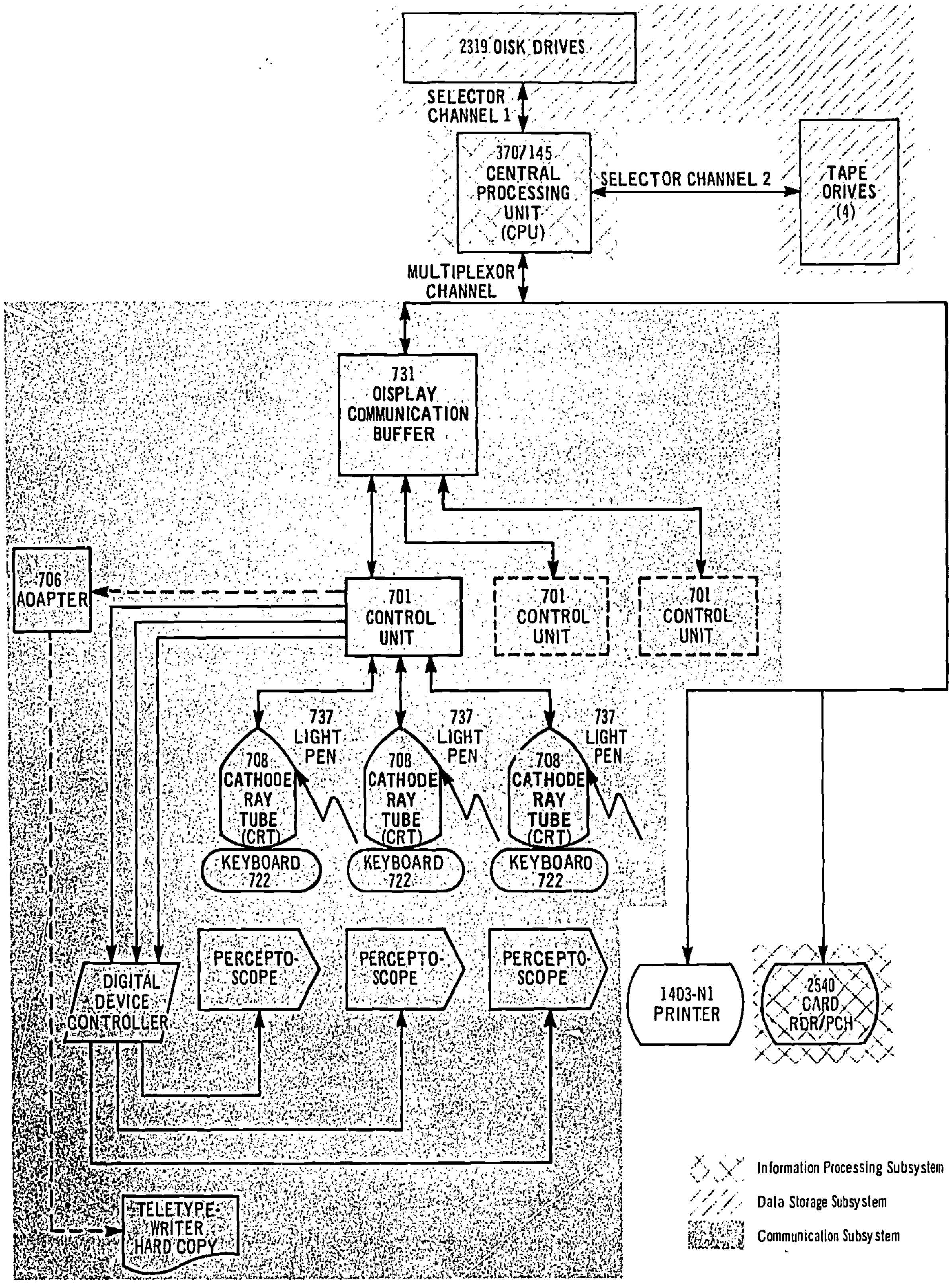


Figure 1

Section 3

OVERVIEW OF IMPACT'S SOFTWARE SYSTEM

REAL-TIME VS. OFF-LINE OPERATIONS

Software system components can be distinguished by dividing them between those that are used in "real-time" and those that are used "off-line." In typical real-time operations, a user of the computer inputs information to which the computer immediately reacts. Through a continuous series of user-input/computer-reaction steps a task is performed, a problem is solved, instruction is presented, and so forth. Other terms that are often used to convey meanings similar to "real-time" are "on-line" and "interactive." The real-time software is used to support activities (1) and (2) in Section 2—presenting instruction and implementing instructional material.

"Off-line" operations are those that are performed with no intermediate user/computer interaction. Typically, a job is submitted to the computer and at some future point in time the computer processes the job. Output from off-line operations is usually in the form of readable printout, new or updated computer files, punched cards, and so forth. The term "batch processing" is often used to convey a meaning similar to the term "off-line." Activities (3) and (4) in Section 2—evaluation and administrative function—are supported by the off-line software.

The speed of contemporary computers makes it possible for the computer to monitor several real-time users simultaneously. For example, during the several seconds it usually takes for a user to prepare an input statement for the computer, the computer can process or monitor the activities of several other users. This feature is the key to the potential for using computers in an instructional environment. It permits numerous students to work simultaneously at different tasks, also within different courses. It is even possible to perform some off-line operations at the same time that on-line operations are proceeding. For example, the computer can administer instruction to students while it is processing evaluation data off-line. This is often referred to as foreground/background operation, and is the way that IMPACT's computer software operates.

The simultaneous use of the computer by several users in real-time is known as "time sharing." Time sharing in an instructional environment must be software based. That is, the nature of individualized instruction is such that it usually requires the capability for making a vast number of varied and complex decisions. Large amounts of data must be collected, stored, and monitored concerning system status as well as system and student performance. At present, it is not feasible to perform these functions with hardware alone. Therefore, instructional time sharing is the burden of the system's software.

IMPACT'S REAL-TIME SOFTWARE SYSTEM

The functional architecture for IMPACT's real-time software system is presented in Figure 2, which shows that numerous terminals can be supported by the system. All input and output passes through some portion of the computer operating system. A system monitor serves to interface logically the terminals and the CAI language

Functional Architecture of IMPACT's Real-Time System

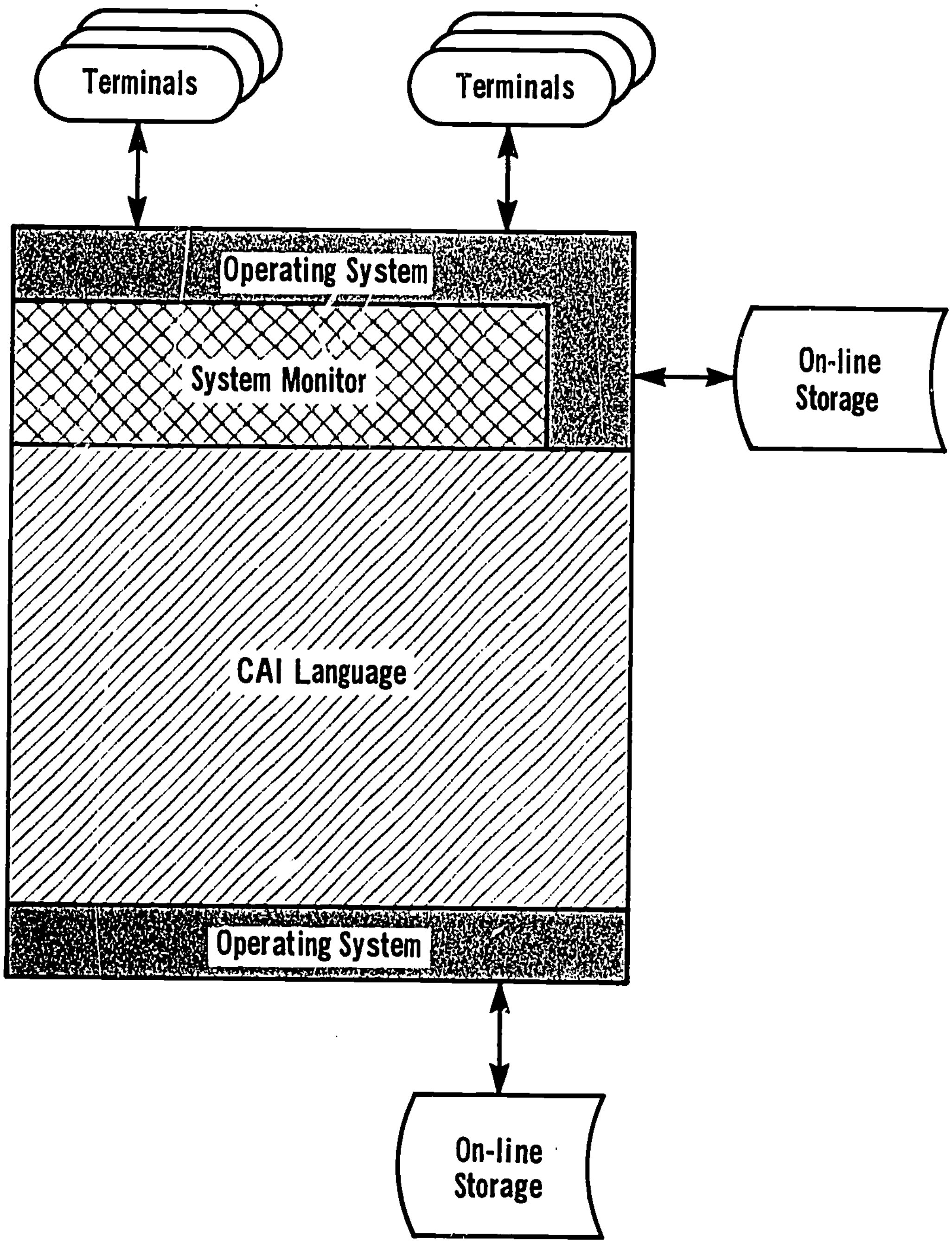


Figure 2

component. The actual monitor, Zeus, is capable of interfacing practically any CAI language that is compatible with the IBM 370 and its terminals. The CAI language component serves in both the student and author modes of operation in order to both present and control instruction, as well as to prepare instructional material.

The actual components in IMPACT's real-time software system are shown in Figure 3. For the sake of completeness, the terminals have been included, even though

IMPACT's Real-Time Software Components

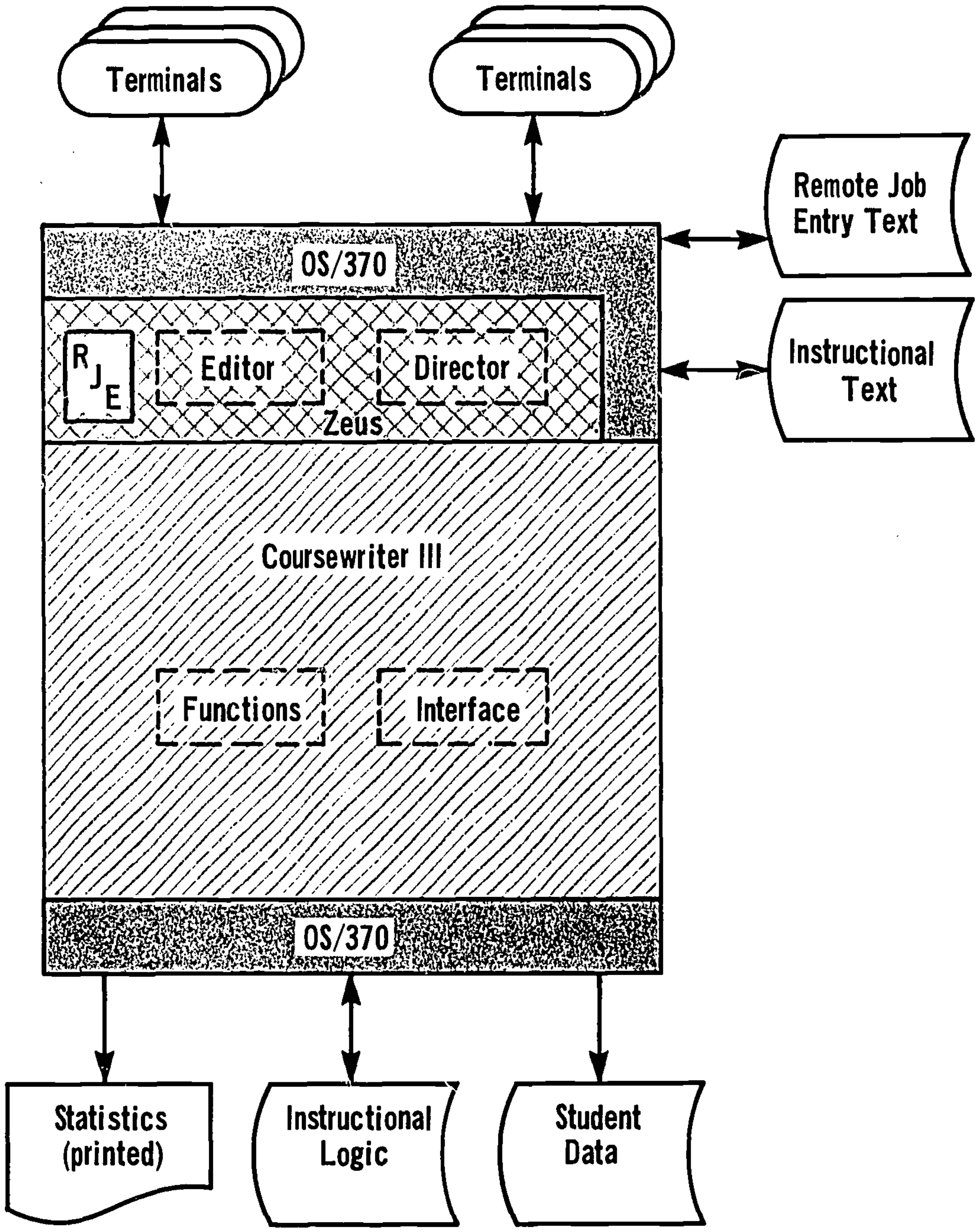


Figure 3

they do not contain any software. The function of each of these components will be described in turn, starting at the top of the figure with the terminals.

Terminals

Each *terminal* consists of a CRT and a random-access, variable speed 16mm motion picture projector that can display single frames. Both of these devices are used to display

information. The terminal also includes a keyboard for inputting information. A lightpen can be used with the CRT in order to relocate the cursor on the CRT. A student using the terminal has instructional content displayed on the CRT and the motion picture projector. A student uses the keyboard for entering responses or comments, or for making information requests. As presently utilized in IMPACT, the information request facility is used for providing definitions of terms for students upon request. An author using the terminal enters commands via the keyboard, uses the CRT and projector for viewing instructional material, and receives system information on the CRT. Currently, IMPACT has nine terminals, but this is not a restriction on the number of terminals the real-time software can support.

The Operating System (OS)

The *Operating System* is IBM's OS/370 and contains the normal input/output routines and device support components contained in any 370 (or 360) Operating System. As shown in Figure 3, all information going to or from the terminals, to or from the disk-stored instructional text, to the printer, to or from the disk-stored instructional logic, or to the student data files passes through some portion of the 370 Operating System.

Zeus

Another element in the figure is *Zeus*, which is a monitor that serves as an interface between Coursewriter and the terminals. Student input to Zeus is either a response, a comment, or an information request. Responses and comments are passed on by Zeus to Coursewriter for analysis or storage. An information request is interpreted by Zeus, and *Director*, a Zeus subprogram, locates and retrieves the appropriate information from the disk. Director also locates and retrieves course text from the disk for display to students and authors on the CRT. Zeus, in turn, transmits the information and text to the appropriate terminal via the OS/370 routines. Author input to Zeus is either an *Editor* command or data to be sent directly to Coursewriter. Editor commands allow an author to create, copy, modify, delete, display, and roll-on displays, feedbacks, and other text (such as definitions). Editor interprets the author commands and Zeus monitors and executes them, using Director as required.

Zeus also receives information directed to the terminals from Coursewriter. Some of the information sent by Coursewriter is transmitted directly to the terminal by Zeus through OS/370. Other information sent by Coursewriter consists of implicit requests to retrieve text from the instructional text file and to transmit this information to the terminal. Zeus interprets the content of all information going to a terminal from Coursewriter. Upon recognizing a text request, Zeus uses Director to interpret the command and to locate and retrieve the required text.

Zeus is also used for system maintenance and Remote Job Entry (RJE). This involves dynamic allocation of input/output buffers and work areas, controlling queues, monitoring the status of system components, and so forth. In effect, Zeus serves as the system executive. It interprets and controls all the information flowing within the system. Zeus activates system components as required and, in general, supervises the system operation. Zeus is a very generalized monitor. It is not restricted to supporting a single CAI component only, but can support concurrently many different components, such as other CAI languages and interactive compilers.

Coursewriter

Coursewriter is a slightly extended version of IBM's standard programming system, Coursewriter III, Version 2, and is used during both authoring and student activities. For

authoring purposes, Coursewriter constructs, edits, or modifies the *Instructional Logic* files. These files contain the Coursewriter commands used to interpret student responses, control branching within the course, and record student data. When a student is using a terminal, Coursewriter maintains the correct portion of the instructional logic in core and interprets it as appropriate. Coursewriter analyzes responses, performs branching within the course, and records data such as internal counters, statistics, and student responses, on tape or disk files. Coursewriter can be used to fetch and execute nonstandard, user-generated *Functions* to assist in the performance of these tasks. IMPACT uses Assembly Language to implement Functions for this purpose.

IMPACT courses are partitioned. Different course partitions can have different prerequisites. The *Interface*, which is a Coursewriter-driven program, keeps track of partition prerequisites as well as an updated version of student attributes. Thus, the Interface contains data through which it can determine what partitions are appropriate for each student at any point in the course. This information can be used to present options to a student concerning the next course partition he might enter.

Coursewriter III, Version 2, has been modified slightly in the IMPACT project to better suit IMPACT's requirements. However, these changes have not affected the basic operational design of Coursewriter.

Instructional Text and Instructional Logic

Finally, the *Instructional Text* and *Instructional Logic* programs are located in separate files on one or more random-access devices. *Student Data* is stored temporarily on random-access files and later on magnetic tapes, while system and student *Statistics* are printed.

Discussion

The most distinguishing features of the IMPACT real-time software system are presented in the following discussion:

(1) The feature that distinguishes IMPACT's system from all standard CAI systems is Zeus, a sophisticated time-sharing monitor that completely controls the real-time operations of the system. Zeus allows several users to be active on the system simultaneously. Zeus can support more than one real-time component and can initiate any off-line program executable in the computer, such as compilers, statistical programs, and user programs in general. (Zeus is not restricted to controlling only Editor, Director, and Coursewriter.)

(2) Another essential feature of the system is that a variety of activities can be under way at the terminals concurrently. This means that it is possible for students and authors to be active in the CAI mode at the same time, while other users (such as programmers) are submitting jobs in the Remote Job Entry mode. Zeus, through Editor, is capable of discriminating between the incoming activity requests and can activate the appropriate program components.

(3) From the point of view of CAI programs, the IMPACT system is unique in that instructional text and instructional logic are stored in separate files. This design facilitates maintenance of the material, in that textual components can be modified without causing major reformulations of the instructional material. For example, if large displays were stored imbedded in the instructional logic and a display were extended, the complete course would have to be re-stored in order to make room for the additional information. In a CRT environment such as IMPACT's, extensive editing is normal for the CRT displays. IMPACT's authoring system is designed so that it is simple to retrieve, modify, and store displays again. This would not be possible unless the instructional text and logic were stored separately. Separate storage has also made it possible to develop the

authoring system in such a way that nontechnical personnel can be used to perform most of the authoring chores.

IMPACT'S OFF-LINE SOFTWARE SYSTEM

IMPACT's off-line software system components are shown in Figure 4. The function of each of these components will be described in this section of the report.

The Operating System

As in real-time software, the off-line software system uses OS/370, IBM's 370 Operating System, that contains the normal input/output routines and device support components of any 370 Operating System. All input/output in the off-line operations passes through some portion of the OS/370.

IDES

IDES (the IMPACT Data Evaluation System), retrieves Student Data generated during real-time and structures working Data files. This is the *Storage and Retrieval* portion of IDES. IDES also provides for the analysis of student-generated data through statistical analysis. IMPACT has two operational, distinct IDES that differ in their storage and retrieval component.

IDES-1, the original version of IDES, performs the Storage and Retrieval function through list processing techniques. A modified SLIP processor is used for this purpose. SLIP is a system that permits the construction, maintenance, and updating of complex list structures (the *Data Files* in IDES-1 are list structures). SLIP also includes routines that can be used to retrieve selective data from lists by searching through lists. These data can be stored in sequential files for subsequent analysis through the use of BMD or other statistical analysis routines, as well as special report generation routines. The *BMD* is the standard Biomedical statistical analysis package developed and distributed by UCLA (6).

The list structuring used in IDES-1 is the most distinguishing feature of the off-line software system. List processing is a technique that is suited uniquely to applications in which it cannot be predicted what data items will be generated. For example, in an educational environment, it cannot always be predicted what questions or even how many times each question will be answered by a student. Therefore, some nonlinear data storage structure often seems to be required for organizing student-generated data, especially when an extremely large amount of data is to be saved.

It was found in Project IMPACT that, for the present, more conventional data storage and retrieval procedures would suffice. Thus, IDES-2 was developed to simplify and streamline the data storage and retrieval procedures. It also relies on the BMD package for data analysis (6). However, IDES-2 contains a more extensive report generation capability than IDES-1.

IDES-2 does not use SLIP in performing the Storage and Retrieval function; it uses standard file maintenance procedures as provided by such standard components of IBM's operating system as PL/1, COBOL, and FORTRAN. IDES-2 is conceptually much simpler than IDES-1, since data are not stored in complex list structures.

FACS

Another distinguishing feature of the off-line system is FACS (File Activity Control System). FACS assists authors in preparing CAI courses. FACS prepares *Reports*

IMPACT's Off-line Software Components

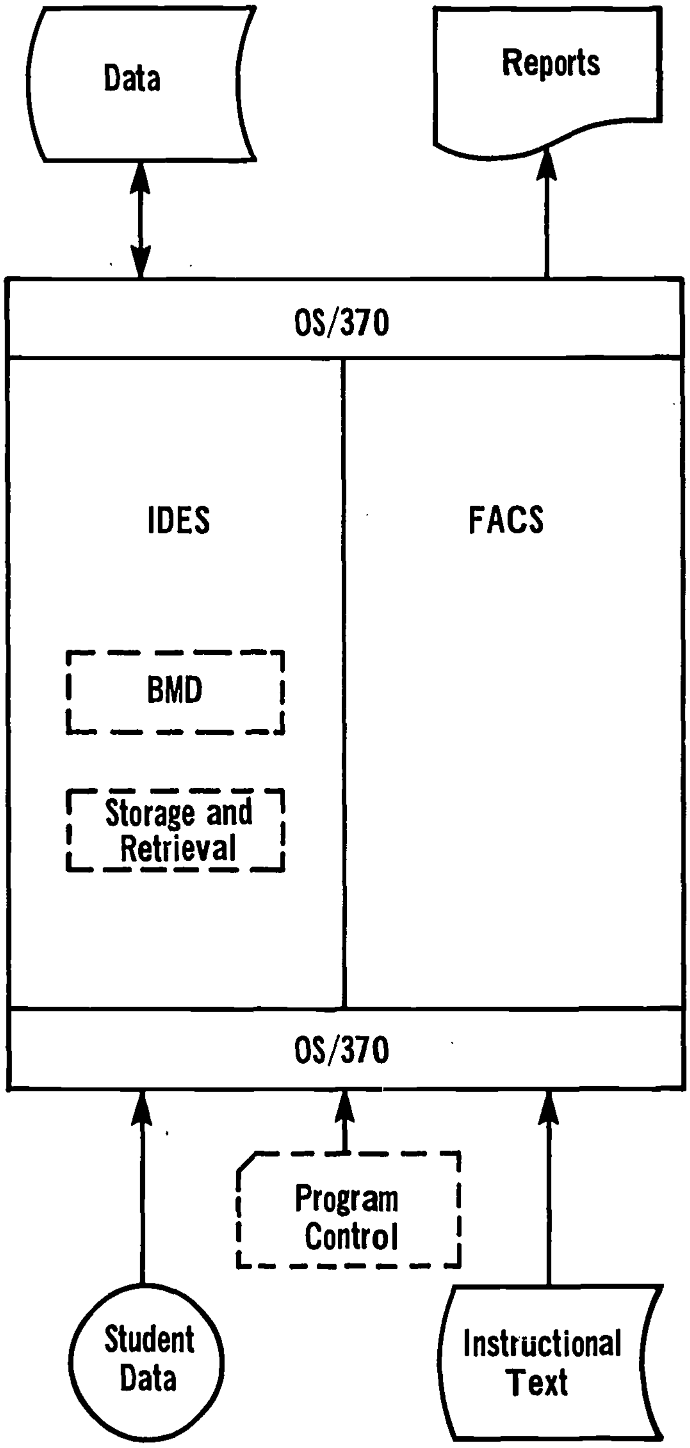


Figure 4

requested by authors concerning the *Instructional Text*. The following are examples of the reports FACS can generate:

(1) A list, by name, of all displays that have been created, revised, or deleted since a specified date.
(2) A list of all displays created by a given author.
(3) A condensed version of a series of related displays.
(4) A printed copy of all or selected displays in the format in which they appear on the CRT screen, along with a coded representation of how they appear in computer memory. Administrative information is also provided, such as name of the display, the author, the date last modified, and the number of characters in the display.
(5) A list, by name, of displays that contain a specified word or series.

The FACS output has a variety of uses such as editing text, scheduling course implementation activities, and contributing to the supervision of the course implementation team. Printouts of replicas of CRT displays provided by FACS are valuable in preparing or documenting course material, and the management information generated by FACS is useful in monitoring file and author activity.

The *Program Control* box in Figure 4 indicates the manner in which programs are initiated and controlled. This input would be in the form of punched cards, internally stored control routines, console input, and so forth.

Section 4

SYSTEM LOGIC

A description of the system operating logic is presented in this section of the report. As in Section 3, the emphasis is on showing how the main components in the software system interact.

Details of the inner logic of the components, such as Zeus, will be presented in other reports (Research Products), in which flowcharts will be used to summarize the salient points of system logic that were made in the discussion in Section 3. Normal flowcharting conventions will be followed, so the diagrams will require little annotation.

REAL-TIME SYSTEM LOGIC

In this section of the present report, real-time and off-line operations are distinguished from one another. An overview of the system logic for real-time operations is presented in Figure 5.

Input to the system from the terminal is either student, author, or remote job user input. Student input is either a response, including special Interface options, a comment, or an information request. Student responses and comments must be processed by Coursewriter while Zeus, through Director, processes information requests. Author input consists of either instructional logic commands to be processed by Coursewriter, or Editor/Director commands or data for processing by Zeus. Remote Job Entry (RJE) input is either commands or expected input that is passed on to the appropriate processing routines by Zeus. (The processing of incoming information is summarized in Figure 6.)

As shown in Figure 5, all input and output in the system use the standard OS/370 input/output processes. Upon receiving input, Zeus checks to see whether it is intended for Editor, Director, or RJE. Input data for Editor, Director, or RJE are processed as appropriate. Editor, Director, and RJE commands are processed using Instructional Text and RJE Files as appropriate. For example, author-input Editor commands might cause some processing to be performed on the Text files. On the other hand, an information request from a student will be recognized as a Director command. When this command is processed the information will be retrieved from the Text files. Similarly, RJE commands will be processed as appropriate. After command processing, any messages intended for the terminal are processed and sent.

Any input that is not intended for Editor, Director, or RJE processing is assumed to be Coursewriter input. This means the input is either a student response including an Interface option, a student comment, or author input for the Instructional Logic Files. Coursewriter processes the input using Instructional Logic Files and stores student data as appropriate. The processing can consist of response analysis or interpreting the Instructional Logic command. Normally Coursewriter responds to the input by either sending a message to the terminal or by erasing the screen. When a message is sent to the terminal, Zeus scans it to see whether a Director command is present. If a Director command is present, Director is invoked causing retrieval from the Instructional Text files. The appropriate message is assembled and sent to the terminal.

Overview of Real-Time System

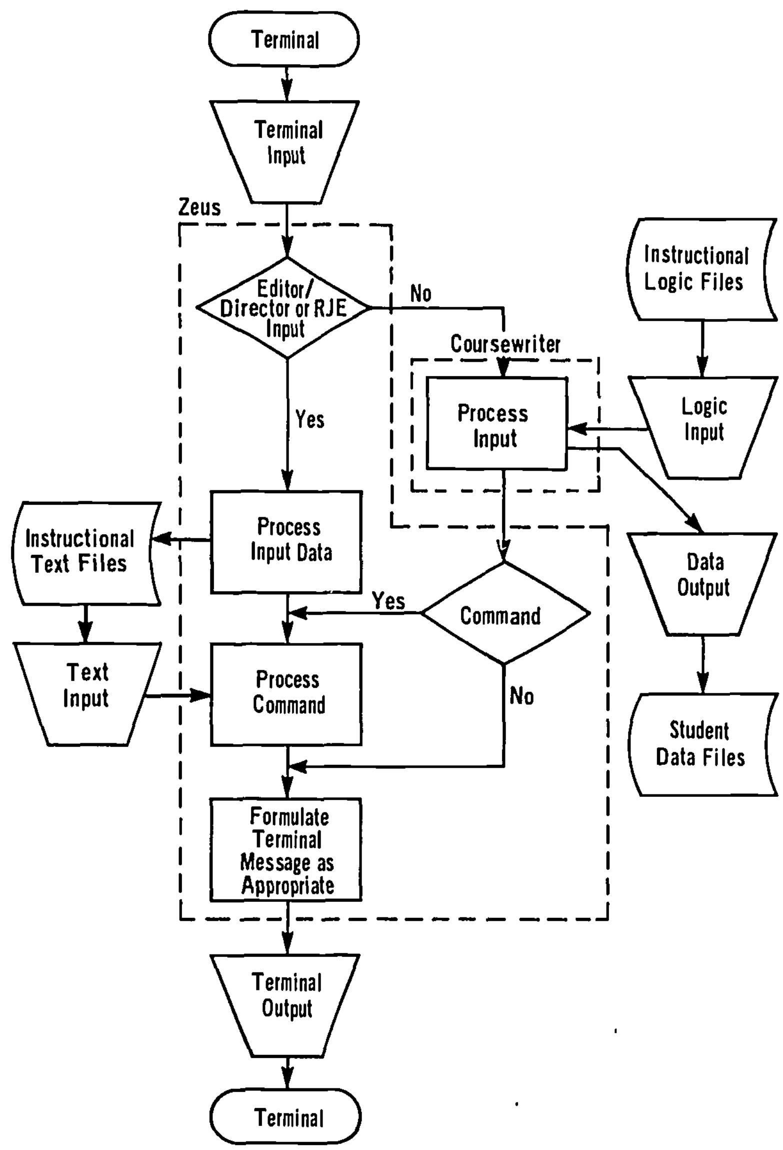


Figure 5

The real-time logic is described in more detail in Figure 7. As shown in Figure 7, Zeus will set up processing before receiving input whenever it is possible to predict what input will be received next. Anticipated input is routed directly to the appropriate processing point in Zeus. For example, when an author wants to create a display from a terminal he first enters the "create display" command with the appropriate display identifier. Zeus interprets this command by preparing to process the next input from the terminal. Zeus erases the CRT and waits for the next input (follow the "no-terminal-output" line in Figure 7). Zeus assumes that the next time data are transmitted from that terminal, the data will consist of the contents of the display being created. Anticipating this input, the next data sent from that terminal are immediately stored in the Instructional Text File. Zeus does not interpret this input, but merely sends them to the appropriate point in the text file for storage.

Input to the Real-Time System

Initiator	Type of Input	Processed by
Student	Response (including special Interface options)	Coursewriter
	Comment	Coursewriter
	Information Request	Zeus/Director
Author	Input to Instructional Logic Files	Coursewriter
	Editor/Director commands and data	Zeus/Editor
Remote Job User	Commands or data	Zeus

Figure 6

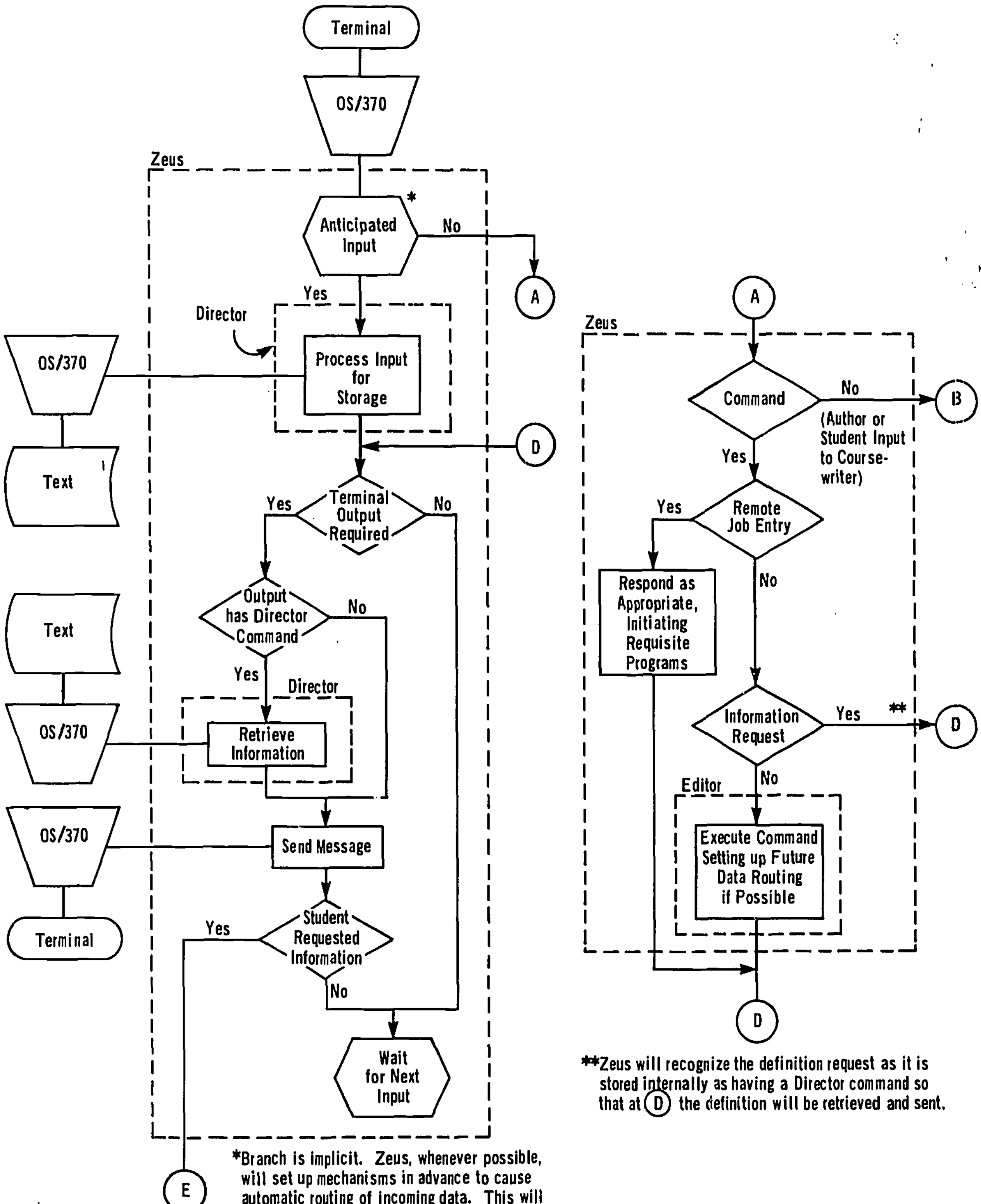
As shown in Figure 7, some anticipated input requires that the system respond with terminal output. *All* output intended for the terminal is scanned by Zeus to see whether it includes a Director command. If the output does contain a Director command, Zeus uses this command to locate the appropriate text in the text file. The text is retrieved and the information is transmitted to the terminal via OS/370 input/output routines. After sending information to the terminal, Zeus checks to see whether student-requested information has just been sent. As mentioned earlier, IMPACT presently uses the information-requesting facility for providing definitions of terms requested by students. If such information has not been sent, Zeus essentially "waits" for the next input from the terminal. Waiting for Zeus is implicit in that Zeus does not actually enter a waiting, idling, or polling routine, but only "pays attention" to a terminal when the terminal requests action from the system.

In a situation when student-requested information has been sent to the terminal, after sending the message control passes from Zeus to Coursewriter (see E in Figure 7). Coursewriter stores a data record containing the information request and identifiers indicating the point in the course where the information was requested. If a message must be sent to the student, Coursewriter sends the information to Zeus. If a message is not required, Coursewriter simply returns control to Zeus. The terminal output processing section of Zeus (see D Figure 7) is re-entered by the system at this point. The processing in this section of Zeus has already been described. In the situation being considered, after student-requested information is sent, requisite text is retrieved from the text files. After sending any required messages, the system is ready to receive new input from the terminal that initiated processing activities through the information request.

Now consider input to the real-time system that is not anticipated (see A in Figure 7). Zeus checks to see whether the input is a command. If it is a command, Zeus then checks to see whether it is an RJE command. If it is an RJE command, Zeus responds by initiating the appropriate RJE routines, programs, and so forth. Zeus then branches to its terminal output subsection for processing required terminal output (see D Figure 7).

If the unanticipated input is a command but not an RJE command, Zeus interprets information requests as commands and hence an information request will be isolated at this point. Zeus handles an information request by branching to its terminal output subsection (D in Figure 7). Zeus will recognize that text retrieval is required. Zeus will retrieve the information and send it to the student. Having done so, control passes to

Real-Time System Logic



**Zeus will recognize the definition request as it is stored internally as having a Director command so that at (D) the definition will be retrieved and sent.

*Branch is implicit. Zeus, whenever possible, will set up mechanisms in advance to cause automatic routing of incoming data. This will be done whenever Zeus can anticipate the nature of the next data to be incoming.

Figure 7 *(Continued)*

Real-Time System Logic *(Continued)*

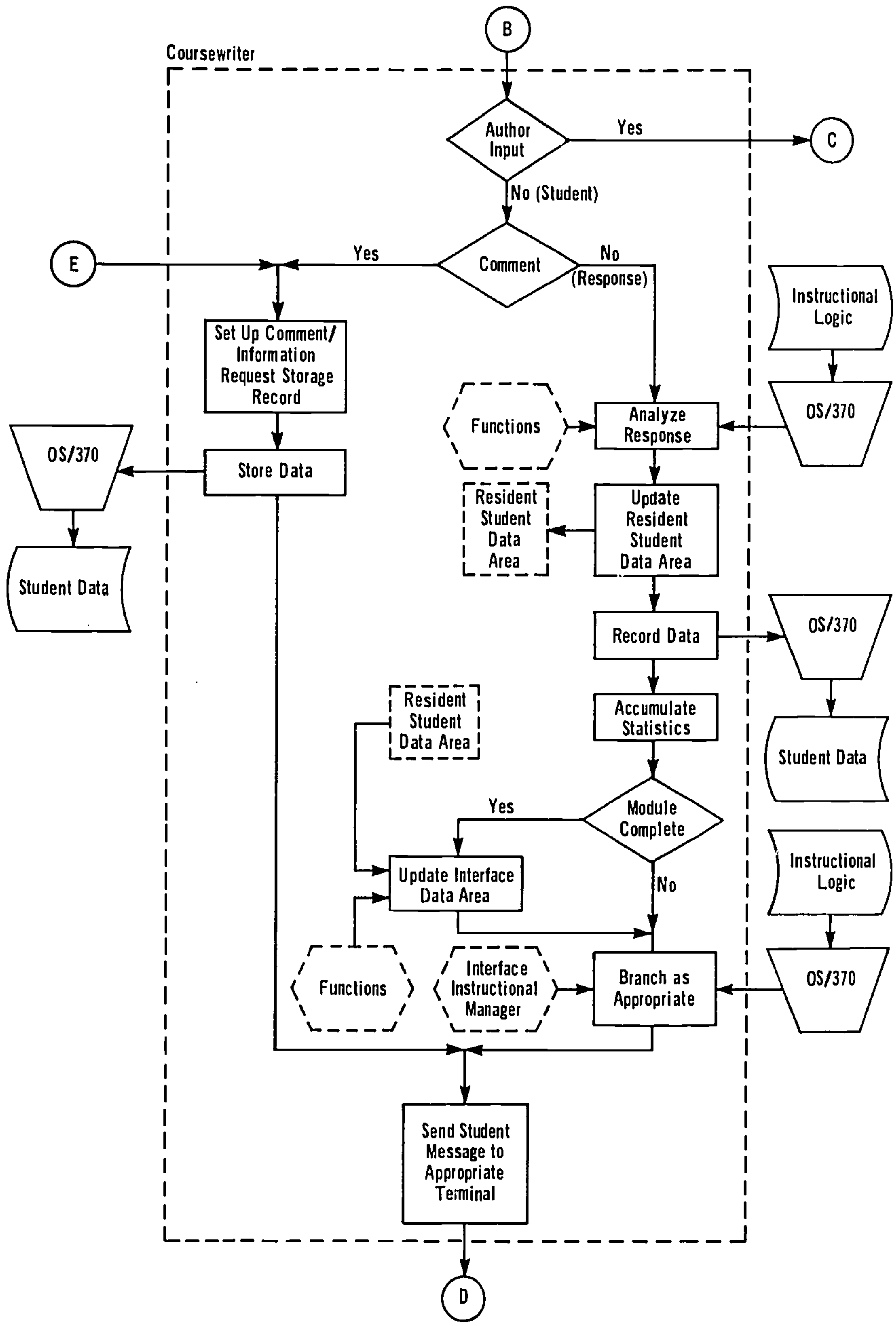


Figure 7 *(Continued)*

Real-Time System Logic *(Continued)*

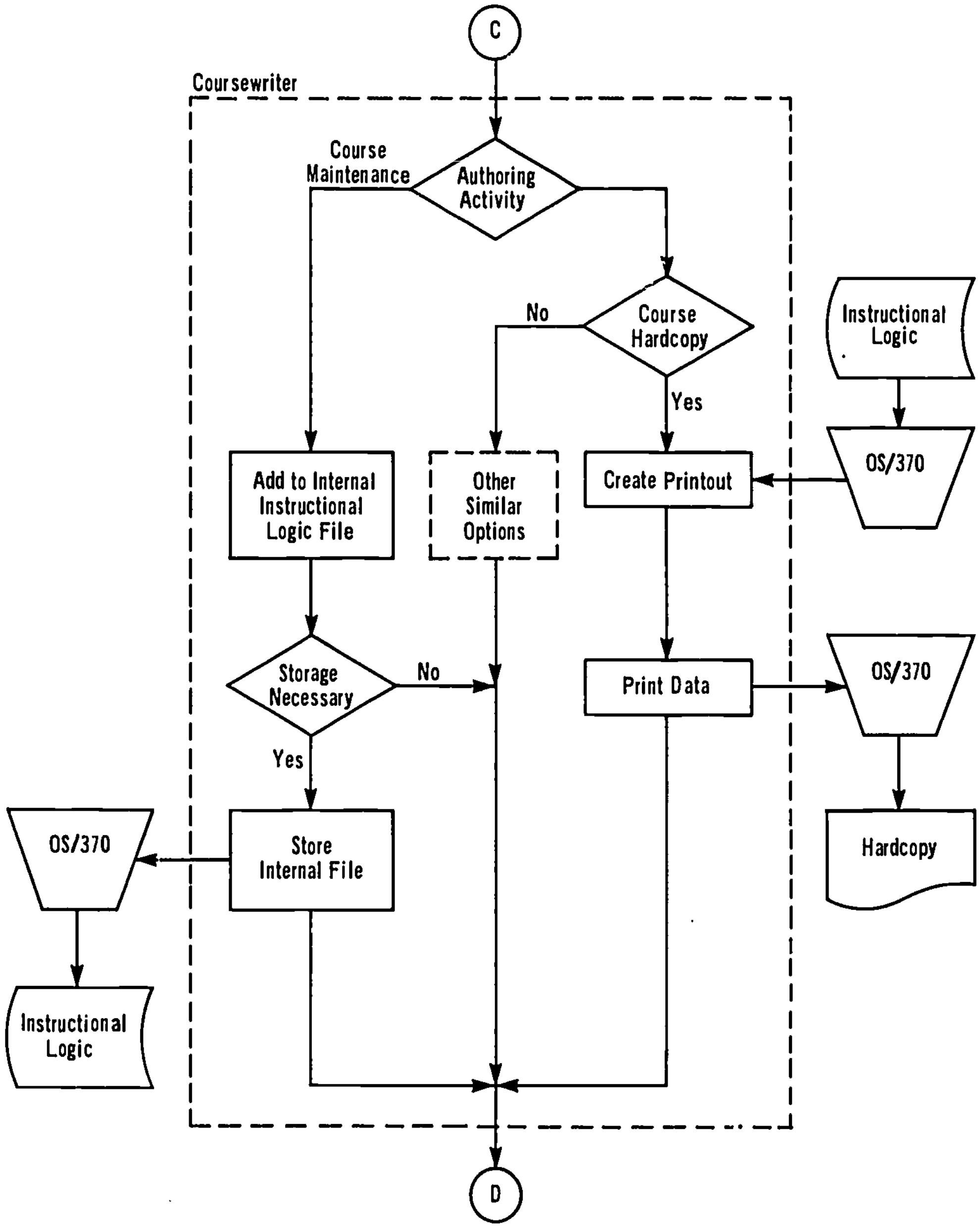


Figure 7

Coursewriter for data storage and then back to the terminal output section of Zeus as described above.

Any unanticipated input that is a command, but not an RJE command or an information request (command), must be an author command. As described in Section 2, an author can create, delete, edit, copy, or modify text. The first input in such a process consists of a command. Through Editor, Zeus interprets the command and sets up the appropriate routing mechanisms for possible subsequent data input. At this point, control

returns to the terminal output section of Zeus (D in Figure 7). If no terminal output is required, Zeus waits for the next input from the terminal, routing it as appropriate.

Now consider unanticipated input from the terminal that is not a command (see B in Figure 7). This input will be either a student response or author input to the Instructional Logic files. A student response must be analyzed by the system. Coursewriter Functions and the Instructional Logic files are often used in this process, as shown in Figure 7. For every response, Coursewriter updates the resident student data area and records student data in the Student Data files. After accumulating required statistics, Coursewriter checks to see whether the student has completed a module. If so, the interface data area is updated. Coursewriter then branches to the next instructional unit. The Instructional Logic files as well as the Interface might play a role in executing this branch operation. Any required student messages are passed on to Zeus for processing by Zeus' terminal output section (see D Figure 7). At this point, Zeus will recognize Director commands in the message and Zeus (Director) will interpret such commands appropriately in order to retrieve instructional text and send it to the student.

Comments entered by the student are distinguished from student responses. Comments are merely stored along with data describing the point at which the comment was made. After recording the data, control returns to Zeus' terminal output subsection (D in Figure 7).

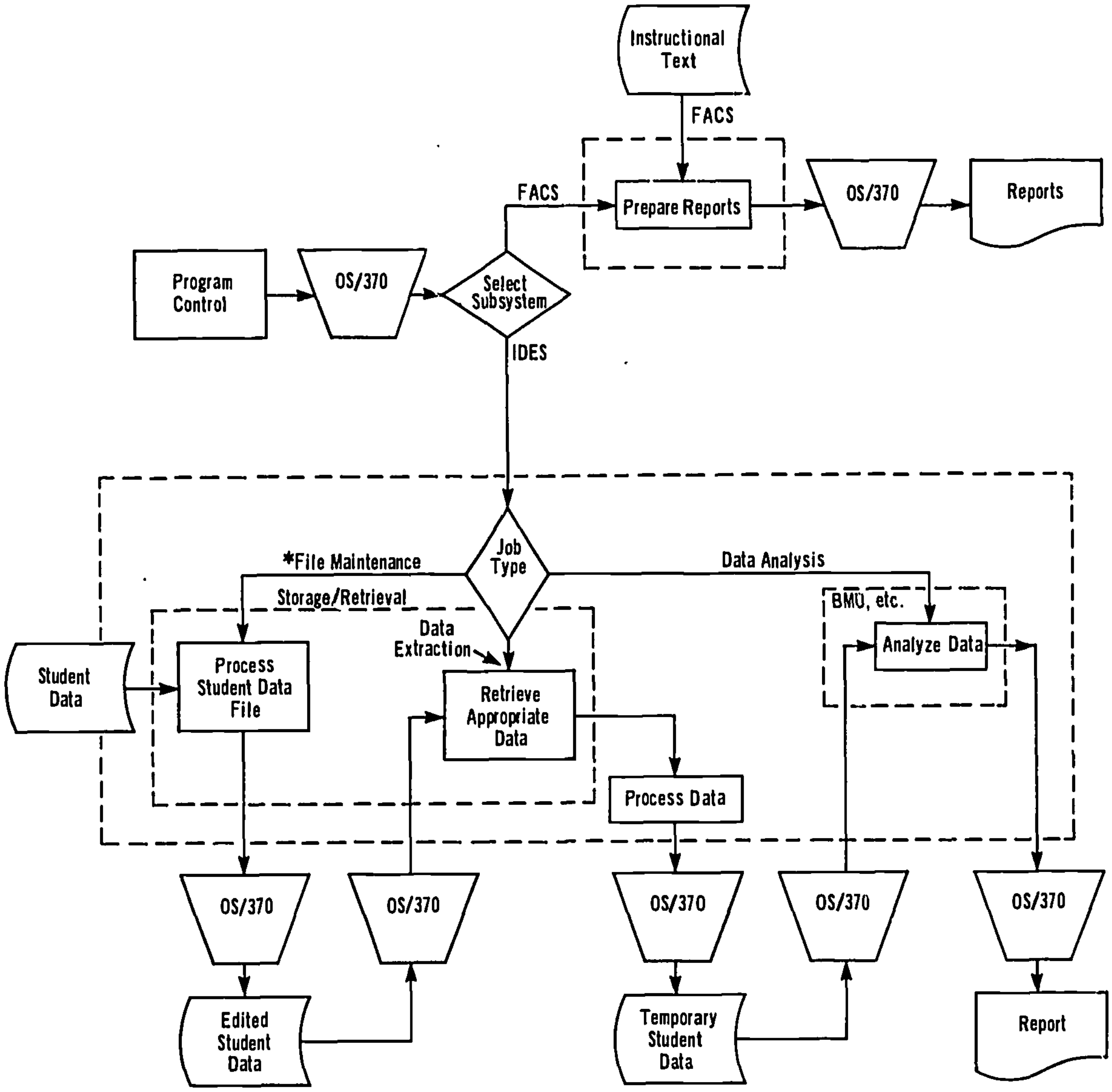
Now consider unanticipated input that consists of authoring data (C in Figure 7). The author might be performing course maintenance, such as entering Coursewriter commands. The commands are stored internally; when the storage buffers are full they are stored in the Instructional Logic files. Other activities are options such as printing segments of courses and printing complete courses. After processing author input, control always returns to the terminal output subsection of Zeus (D in Figure 7).

OFF-LINE SYSTEM LOGIC

The off-line system logic is diagramed in Figure 8. Program control is initiated by punched cards. The system selects either the FACS or IDES subsystem for processing. FACS prepares reports from the Instructional Text files and prints edited hardcopy for subsequent use by authors.

When IDES is initiated, three types of activities can result, as shown in Figure 8 at A . The Storage/Retrieval subsection of IDES processes the Student Data file that has been generated on-line during the course. As a result, relevant data are isolated in the file and an edited file is produced for subsequent processing. The data extraction process causes selected data to be extracted from the edited data file resulting in a temporary student data file. The data analysis activity causes the Temporary Student Data file to be processed by analysis routines such as the BMD. Statistical reports are generated as a result of this activity.

Off-line System Logic



*Performed by Slip in IDES-1

Figure 8

Section 5

SYSTEM REQUIREMENTS

This section contains a brief summary of the requirements a computer system must meet in order for the IMPACT CAI software system to run on that computer. The requirements are minimal in that they constitute the minimum compatible computer configuration.

The first requirement is that the computer be an IBM 360 or 370. The 370 must operate under OS (PCP, MFT, or MVT). The computer must include some form of random access storage (IMPACT uses an IBM 2319 but an IBM 2311 or 3330 disk or 2305 drum would suffice) with the only constraint being the amount of storage required for courses and text. It is possible that the system could be modified to run on a 370 under a different operating system, such as DOS. It is also possible that tape drives could be substituted for the direct access storage. However, such deviations would be major departures from IMPACT's architectural philosophy. Extensive system modification would be required in such cases and some components would probably have to be abandoned.

The most crucial requirement for an alternate computer configuration involves the amount of main memory that is available. The main memory requirements for the system components are summarized in Figure 9, in which both real-time and off-line operations requirements are presented.

Minimum Memory Requirements for System Components

Environment	Component	Minimum Required Main Memory in K* Bytes
Real-Time	Zeus	35 K
	Coursewriter	60 K
	Operating System	19 K
	Total	114 K
Off-line	IDES	90 K
	FACS	90 K
	BMD	90 K (some 200 K requirements)

*K = 1024 bytes

Figure 9

Section 6

CONCLUSIONS

In the foregoing sections, an overview of IMPACT's CAI software subsystem has been presented. The primary components of the subsystem are Zeus, Coursewriter with Functions, the Interface, IDES, and FACS. Research Products in a series of software documents will provide the specific details of each of these components—thus, there is a document describing each of the real-time subsystem components. Documents describing the primary off-line subsystem components are presently in preparation.

As the preceding sections indicate, IMPACT's software system was developed by relying as much as possible on existing software components. Although the real-time portion of the system draws upon the standard IBM Coursewriter-III program, Zeus and the Interface were developed at IMPACT in order to meet specific requirements that Coursewriter could not satisfy. These requirements are based primarily on the fact that IMPACT utilizes a CRT (and, in fact, a Sanders 720).

The real-time components developed by IMPACT are general in that they can be applied in CAI environments that do not use the same CRT. With minimal changes, these components could be used with a range of terminal configurations in other CAI environments. The off-line portion of the software subsystem also relies on existing software components as much as possible. In particular, by using the BMD programs, a great deal of duplicate computer programming has been avoided. The IDES and FACS components were developed to meet specific IMPACT requirements. However, these components are also of general interest since they could be modified readily in order to be applicable in other CAI environments.

All software components developed by IMPACT make use of procedures and methods that are state-of-the-art from a programming point of view. For example, Zeus is a sophisticated time-sharing system that performs such complex tasks as memory allocation and internal management. The methods used in performing functions such as these are used throughout the industry and are recognized as being optimal. Thus, IMPACT's software subsystem makes use of methods and programs that are recognized as providing optimal solutions to the problems being solved.

This does not imply that the total IMPACT CAI system is the optimal operational CAI system available (or even possible). On the contrary, as described in IMPACT's Technical Development Plan, the IMPACT CAI system is being developed through a continuing effort; the software subsystem described herein is a step in that continuing process. In fact, the software subsystem described here can be seen as a portion of the first IMPACT CAI system—IMPACT-A (for an exact description of IMPACT-A and IMPACT-B, see reference 3). As described in the Technical Development Plan, IMPACT-A will become IMPACT-B in the ongoing research and development effort.

The software subsystem described above is now operating. It provides a concrete starting point from which the orderly development of more efficient and sophisticated software components are evolving. An example of this evolution can be seen in the development of IDES-2 from IDES-1. The actual direction and the specific results of this evolutionary process cannot be predicted in detail. It can only be said that future developments will result in concrete contributions to the advancement of the state of the art for CAI.

This overview provides a description of the components of the current IMPACT software subsystem. It shows a picture of development based as much as possible on existing software components. The real-time components have been developed to be general and applicable, with minimal changes, to CAI environments that do not use the same CRT. The off-line components rely as much as possible on existing software (e.g., the UCLA Bio-Med statistical package, 6).

Other data management components were developed to meet specific IMPACT requirements while retaining general characteristics also useful in other CAI (and some non-CAI) environments. The software subsystem is designed to progress toward further achievements of CAI.

LITERATURE CITED

1. Seidel, Robert J. *Current Status of Computer-Administered Instruction Work Under Project IMPACT*, Presentation at CONARC Training Workshop, Fort Gordon, Ga., October 1971; HumRRO Professional Paper 18-72, July 1972.
2. Seidel, Robert J. and the IMPACT Staff. *Project IMPACT: Computer-Administered Instruction Concepts and Initial Development*, HumRRO Technical Report 69-3, March 1969.
3. The IMPACT Staff. *Project IMPACT—Computer-Administered Instruction: Description of the Hardware/Software Subsystem*, HumRRO Technical Report 70-22, December 1970.
4. The IMPACT Staff. *Project IMPACT—Computer-Administered Instruction: Functions for the Coursewriter III Language*, HumRRO Research Product RBP-D1-71-2, June 1971.
5. The IMPACT Staff. *Project IMPACT—Computer-Administered Instruction: Preparing and Managing the Content of Instruction, IMPACT Text-Handling Subsystem*, HumRRO Technical Report 71-21, September 1971.
6. Dixon, W.J. (ed.) *BMD Biomedical Computer Programs*, University of California Publications in Automatic Computation No. 2, University of California Press, Berkeley and Los Angeles, Calif., 1970.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Human Resources Research Organization (HumRRO) 300 North Washington Street Alexandria, Va. 22314	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Project IMPACT Software Documentation: Overview of the Computer-Administered Instruction Subsystem

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Technical Report

5. AUTHOR(S) *(First name, middle initial, last name)*

John Stelzer and Jean Garneau

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1972	36	6

8a. CONTRACT OR GRANT NO. DAHC 19-73-C-0004 b. PROJECT NO. CDA Proj 2Q063101D734 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) HumRRO Technical Report 72-21 9b. OTHER REPORT NO.(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
HumRRO Division No. 1 (System Operations), Alexandria, Va.; Work Unit IMPACT	Office, Chief of Research and Development Department of the Army Washington, D.C. 20310

13. ABSTRACT

This overview of the IMPACT Computer-Administered Instruction Software Subsystem describes the rationale and motivation for the development of the software. A series of Research Products, available through information retrieval systems, will completely document the technical details of the software. In this overview, software subsystem components are identified and their interactions are described. The system is described in terms of real-time vs. off-line operations. Flowcharts are provided to show the general pattern of information processing within the system.

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer-Administered Instruction (CAI)						
Cathode Ray Tube						
Computer						
Coursewriter						
Instructional Logic						
Instructional Text						
Off-line						
Operating System						
Real-time						
Time Sharing						

Unclassified

Security Classification

DISTRIBUTION LIST

2	DIR DASO MANPOWER IPPEGRI
1	CO ARMY TRANS RES COMD FT EUSTIS ATTN TECH LIB
1	DIR HEL APG MD
2	ENGNR PSYCHOL LAB PIONEERING RES DIV ARMY NATICK LABS NATICK MASS
4	TECH LIB ARMY NATICK LABS NATICK MASS
1	REDSTONE SCIENTIFIC INFO CTR US ARMY MSL COMD ATTN CHF DOC SEC ALA
1	CO FT HUACHUCA SPT COMD USA ATTN TECH REF LIB
1	SIXTH USA LIB DEPOT BLDG M 13 14 PRES OF SAN FRAN
1	PLNS OFCR PSYCH HOQTRES USACDCEC FT ORD
5	CG FT ORD ATTN G3 TNG DIV
1	TECH LIB BOX 22 USACDC EXPERIMENTATION COMD FT ORD
1	HUMAN FACTORS TEST DIV (ADHZ) USAF HOSP EGLIN AFB
1	LIB DEF SUPPLY AGCY CAMERON STA VA
1	DIR OF INTERN TNG USA LOG MGT CTR FT LEE
3	CO USA TNG CTR (FA) ATTN AKPSITC-TT FT SILL
1	CG USA INF CTR ATTN AJIGT-T FT BENNING
3	LIB ARMY WAR COLL CARLISLE BKS
1	COMDT COMD + GEN STAFF CO FT LEAVENWORTH-ATTN ARCHIVES
1	US MILIT ACAD WEST POINT ATTN LIB
1	COMDT ARMY AVN SCH ATTN DIR OF INSTR FT RUCKER
2	COMDT ARMY SECUR AGY TNG CTR + SCH FT DEVENS ATTN LIB
1	STIMSON LIB MED FLD SERV SCH BROOKE ARMY MED CTR FT SAM HOUSTON
10	COMDT THE ARMOR SCH ATTN DDI FT KNOX
1	COMDT ARMY ARMOR SCH FT KNOX ATTN WEAPONS DEPT
1	LIB USA ARMOR SCH FT KNOX
1	COMDT ARMY CHEM CORPS SCH FT MCCLELLAN ATTN EDUC ADV
1	COMDT USA FIN SCH ATTN CHF DOC DEV LIT PLN DIV DDDI IND
1	USA FINANCE SCH FT BENJ HARRISON ATTN EDUC ADV
4	COMDT ADJ GEN SCH FT BENJ HARRISON ATTN EDUC ADV
1	COMDT USAIS ATTN EDUC ADV FT BENNING
1	COMDT USAIS ATTN AJIIS-D-EPRO FT BENNING
1	HQ US ARMY ADJ GEN SCH FT BENJ HARRISON ATT COMDT
1	LIB ARMY QM SCH FT LEE
1	COMDT USA QM SCH FT LEE ATTN EDUC ADV
1	COMDT ARMY TRANS SCH FT EUSTIS ATTN EDUC ADV
1	CO USA SEC AGY TNG CTR & SCH ATTN IATEV RSCH ADV FT DEVENS
1	COMDT USA MIL POLICE SCH ATTN PLNS &PROG DDDI FT GORDON
2	COMDT US ARMY SOUTHEASTERN SIG SCH ATTN EDUC ADV FT GORDON
1	COMDT USA AD SCH ATTN DDI FT BLISS
1	CO USA DAD CTR & SCH OFC OF OPS ATTN AMBN-O APG MD
3	CG USA FLD ARTY CTR & FT SILL ATTN AVN OFCR
1	COMDT DEF INTELL SCH ATTN SIGAS DEPT
1	COMDT ARMED FORCES STAFF COLL NORFOLK
1	COMDT USA SIG CTR & SCH ATTN DDI FT MONMOUTH
1	COMDT JUDGE ADVOCATE GENERALS SCH U OF VA
1	USA AVN SCH ELEMENT OFC OF DIR OF INSTR ATTN EDUC ADV GA
1	EDUC CONSLT ARMY MILIT POLICE SCH FT GORDON
6	COMDT USA ENGR SCH ATTN EDUC ADV AHBBES-EA FT BELVOIR
2	COMDT USA SCH EUROPE ATTN EDUC ADV APO 09172 NY
1	OFC OF DOCTRINE DEV LIT & PLNS USA ARMOR SCH ATTN AHBAAS-DM
1	COMDT ARMY AVN SCH FT RUCKER ATTN EDUC ADV
1	DIR OF INSTR US MIL ACAD WEST POINT NY
1	DIR OF MILIT INSTR US MILIT ACAD WEST POINT
2	COMDT USA MSL & MUN CTR & SCH ATTN CHF OFC OF OPS REDSTONE ARSNL
2	COMDT US WAC SCH US WAC CTR ATTN AJMCT FT MCCLELLAN
1	CO USA INTELL CTR & SCH ATTN DIR OF ACADEMIC OPS FT HUACHUCA
1	CO USA INTELL CTR & SCH ATTN DIR OF DOC & LIT FT HUACHUCA
1	COMDT USA C&GSC OFC OF CHF OF RESIDENT INSTR FT LEAVENWORTH
1	COMDT USA CA SCH ATTN DDI FT BRAGG
1	COMDT USA CA SCH ATTN EDUC ADV FT BRAGG
1	COMDT USA CA SCH ATTN LIB FT BRAGG
10	COMDT USA INST FOR MIL ASSIST ATTN DDI FT BRAGG
8	COMDT USA FLD ARTY SCH ATTN DDI FT SILL
1	COMDT USA ARTY SCH ATTN EDUC SERVICES DIV FT SILL
1	COMDT USA ARTY SCH ATTN EDUC ADV FT SILL
1	COMDT USA TRANS SCH ATTN DIR OF DOC & LIT FT EUSTIS
1	COMDT USA TRANS SCH ATTN LIB FT EUSTIS
1	USA INST FOR MIL ASST ATTN EDUC ADV FT BRAGG
1	COMDT ARMY QM SCH OFC DIR OF NONRESID ACTVY ATTN TNG MEDIA DIV VA
1	COMDT USA ARTY SCH ATTN LIB FT SILL
1	DIR OF GRAD STUD & RSCH ATTN BEHAV SCI REP USAC&GSC
1	DIR OF PERS STUDIES & RSCH DCSPER DA WASH DC
2	ACSFOR DA ATTN CHF TNG DIV WASH DC
1	HQ ARMY MAT COMD R+D DRCTE ATTN AMCRD-RC
1	US ARMY BEHAVIOR & SYS RSCH LAB ATTNCRD-AR ARL VA
12	ADMIN DDC ATTN: TCA (HEALY) CAMERON STA ALEX., VA. 22314
1	CHF OF R+D DA ATTN CHF TECH + INDSTR LIAISON OFC
1	USA BEHAVIOR & SYS RSCH LAB ATTN CRD-AIC ARL VA
1	COMDT USA CBT SURVEIL SCH & TNG CTR ATT ED ADV FT HUACHUCA
1	COMDT USA CBT SURVEIL SCH & TNG CTR ATTN ORG DOC & NEW EQUIP ARIZ
2	TNG & DEVEL DIV DCSPERS
1	USA LIB DIV-TAGO ATTN ASDIRS
1	CG CONARC ATTN COL E M HUDAK ATIT-SA FT MONROE
15	CG CONARC ATTN ATIT-STM FT MONROE
1	CHF USA AD HRU FT BLISS
1	CHF USA ARMOR HRU FT KNOX
1	CHF USA AVN HRU FT RUCKER
1	CHF USA INF HRU FT BENNING
1	CHF USA TNG CTR HRU PRES OF MONTEREY
1	CG 8TH INF DIV ATTN ACOFS G2 APO NY 09111
1	RSCH CONTRACTS & GRANTS BR ARO
1	BESD ARO OFC CHF OF R&D WASH DC
1	CHF OF R&D DA ATTN SCI INFO BR RSCH SPT DIV WASH DC
5	TECH LIB PERS 11B BUR OF NAV PERS ARL ANNEX
3	DIR PERS RES DIV BUR OF NAV PERS
1	ENGNR PSYCHOL BR ONR CODE 455 ATTN ASST HEAD WASH DC
2	CO FLT TNG CTR NAV BASE NEWPORT
1	CO FLEET TNG CTR US NAV STA SAN DIEGO
1	PRES NAV WAR COLL NEWPORT ATTN MAHAN LIB
1	CHF OF NAVL RSCH PERS & TNG BR (CODE 458) ARL VA
1	CHF OF NAV RES ATTN HEAD GP PSYCHOL BR CODE 452
1	DIR US NAV RES LAB ATTN CODE 5120
1	DIR NAVAL RSCH LAB ATTN LIB CODE 2029 WASH DC
1	CHF OF NAV AIR TNG TNG RES DEPT NAV AIR STA PENSACOLA
1	DIR PERS RES LAB NAV PERS PROGRAM SUPPORT ACTIVITY WASH NAV YD
1	DIR MARINE CORPS INST ATTN EVAL UNIT
1	CHF OFCR PERS RES + REVIEW BR COAST GUARD HQ
1	SUPT US COAST GUARD ACAD NEW LONDON CONN
1	AIR TNG COMD/XPT RANDOLPH AFB
1	TECH DIR TECH TNG DIV(HRO) AFHRL LOWRY AFB COLO
1	CHF SCI DIV DRCTE SCI + TECH DCS R+D HQ AIR FORCE AFRSTA
1	CHF OF PERS RES BR DRCTE OF CIVILIAN PERS DCS-PERS HQ AIR FORCE
1	CHF ANAL DIV (AFPDPL (R) DIR OF PERSONNEL PLANNING HQS USAF
1	AFHRL/TT ATTN CAPT W S SELLMAN LOWRY AFB
2	AFHRL (HRT) WRIGHT-PATTERSON AFB
1	USAFA DIR OF THE LIB USAF ACAD COLO
2	CO HUMAN RESOURCES LAB BROOKS AFB
1	AFHRL (FT) WILLIAMS AFB ARIZ
1	PSYCHOBIOLOGY PROG NATL SCI FOUND
1	DIR NATL SECUR AGY FT GEO G MEADE ATTN DIR OF TNG
3	CIA ATTN CRS/ADD STANDARD DIST
1	SYS EVAL DIV RES DIRECTORATE DOD-DCD PENTAGON
1	DEPT OF STATE BUR OF INTEL + RES EXTERNAL RES STAFF
1	SCI INFO EXCH WASHINGTON
2	CHF MGT & GEN TNG DIV TR 200 FAA WASH DC
1	BUR OF RES & ENGR US POST OFC DEPT ATTN CHF HUMAN FACTORS BR
1	EDUC MEDIA BR OE HEW ATTN T D CLEMENS
2	ERIC OE WASH DC
1	CONSOL FED LAW ENFORCEMENT TNG CTR WASH DC
2	RAC ATTN LIB MCLEAN VA
1	RAND CORP WASHINGTON ATTN LIB
1	MITRE CORP BEDFORD MASS ATTN LIB
2	LEARNING R&D CTR U OF PITTS ATTN DIR
1	HUMAN SCI RES INC MCLEAN VA
1	IDA RSCH & ENG SUPT DIV ARL VA
1	SCI & TECH DIV IDA ARL VA
1	DIR CTR FOR RES ON LEARNING + TEACHING U OF MICH
1	EDITOR TNG RES ABSTR AMER SOC OF TNG DIRS U OF TENN
1	AMER INSTS FOR RSCH ATTN LIBN PA
3	MATRIX RSCH CO FALLS CHURCH VA
1	DR GEORGE T HAUTY CHMN DEPT OF PSYCHOL U OF DEL
1	HEAD DEPT OF PSYCHOL UNIV OF SC COLUMBIA
1	AMER PSYCHOL ASSOC WASHINGTON ATTN PSYCHOL ABSTR
1	NO ILL U HEAD DEPT OF PSYCHOL
1	GEORGIA INST OF TECH DIR SCH OF PSYCHOL
1	LIFE SCI INC HURST TEXAS ATTN W G MATHENY
1	AMER BEHAV SCI CALIF
1	SO ILLINOIS U DEPT OF PSYCHOL
1	DR E FOULKE DEPT OF PSYCH UNIV OF LOUISVILLE
1	DR C HELM DEPT EDUC PSYCH CITY U OF NY
1	PSYCHOL LIB HARVARD UNIV CAMBRIDGE
1	CATHOLIC U LIB EDUC & PSYCHOL LIB WASH DC